AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method comprising:

generating client message digests at a client, the client message digests corresponding to client files stored on the client, wherein each client message digest corresponds to each client file on the client, wherein the client message digests uniquely identify contents of the client files via unique fingerprints corresponding to the client files, wherein the unique fingerprints are generated based on the contents of the client files by performing a cryptographic hash of the contents of the client files, wherein the client files are cataloged by the client message digests;

generating ~~server~~ repository message digests corresponding to ~~server~~ repository files, each ~~server~~ repository message digest corresponding to a ~~server~~ repository file on a ~~server~~ repository, wherein the ~~server~~ repository is coupled to the client over a network;

prior to synchronizing the client files with the ~~server~~ repository files, matching client file contents from the client message digests with ~~server~~ repository file contents from the ~~server~~ repository message digests to determine whether the client files and the ~~server~~ repository files are to be synchronized;

synchronizing the client files and the ~~server~~ repository ~~files,~~ files if the client file contents and the ~~server~~ repository file contents do not match, wherein the synchronizing of the client files and the ~~server~~ repository files further includes marking those ~~un-matching~~unmatched files of the client files and the ~~server~~ repository files that remain unsynchronized to be copied to ~~a~~ the repository ~~for matching~~ to be synchronized at a later time;

performing a post-synchronization match of the client message digests with the ~~server~~ repository message digests and, if the client message digests do not match the ~~server~~ repository message digests, detecting one or more client files corresponding to one or more unmatched client message digests, and tagging the one or more client files; and

re-synchronizing the client files and the ~~server~~ repository files, the re-synchronization including copying the one or more client files to the ~~server~~repository such that the client message digests and the ~~server~~ repository message digests are matched.

2. (Currently Amended) The method of claim 1, wherein the synchronizing of the client files and the ~~server~~repository files comprises adding client file contents that are missing on the ~~server~~repository~~ to the server~~.

Claims 3 - 6 (Cancelled)

7. (Previously Presented) The method of claim 1, further comprising combining the client message digests into a single client message digest.

Claims 8-9 (Cancelled)

10. (Currently Amended) A system comprising:

a storage medium; and

a processor coupled with the storage medium, the processor to:

generate client message digests at a client, the client message digests corresponding to client files stored on the client, wherein each client message digest corresponds to each client file on the client, wherein the client message digests uniquely identify contents of the client files via unique fingerprints corresponding to the client files, wherein the unique fingerprints are generated based on the contents of the client files, wherein the client files are cataloged by the client message digests;

generate ~~server~~repository message digests corresponding to ~~server~~repository files, each ~~server~~repository message digest corresponding to a ~~server~~repository file on a ~~server~~repository, wherein the ~~server~~repository is coupled to the client over a network;

prior to synchronizing the client files with the ~~server~~repository files, match client file contents from the client message digests with ~~server~~the repository file contents from the ~~server~~repository message digests to determine whether the client files and the ~~server~~repository files are to be synchronized;

synchronize the client files and the ~~server~~repository ~~files,~~ files if the client file contents and the ~~server~~repository file contents do not match, wherein the synchronizing of the client files and the ~~server~~repository files further includes marking ~~un-matching~~ those unmatched files of the client files and the ~~server~~repository files that remain unsynchronized to be copied to ~~a~~ the repository ~~for matching~~ to be synchronized at a later time;

perform a post-synchronization match of the client message digests with the ~~server~~repository message digests and, if the client message digests do not match the ~~server~~repository message digests, detect one or more client files corresponding to one or more unmatched client message digests, and tag the one or more client files; and

re-synchronize the client files and the ~~server~~repository files, the re-synchronization including copying the one or more client files to the

~~server~~repository such that the client message digests and the ~~server~~repository message digests are matched.

11. (Cancelled)

12. (Previously Presented) The system of claim 10, wherein the cryptographic hash comprises 128 to 160 bits.

Claims 13-19 (Cancelled)

20. (Currently Amended) A machine-readable medium comprising instructions which, when executed, cause a machine to:

generate client message digests at a client, the client message digests corresponding to client files stored on the client, wherein each client message digest corresponds to each client file on the client, wherien the client message digests uniquely identify contents of the client files via unique fingerprints corresponding to the client files, wherein the unique fingerprints are generated based on the contents of the client files by performing a cryptographic hash of the contents of the client files, wherein the client files are cataloged by the client message digests;

generate ~~server~~repository message digests corresponding to ~~server~~the repository files, each ~~server~~repository message digest corresponding to a ~~server~~repository file on a ~~server~~repository, wherein the ~~server~~repository is coupled to the client over a network;

prior to synchronizing the client files with the ~~server~~repository files, match client file contents from the client message digests with ~~server~~the repository file contents from the ~~server~~repository message digests to determine whether the client files and the ~~server~~repository files are to be synchronized;

synchronize the client files and the ~~server~~repository ~~files,~~ files if the client file contents and the ~~server~~repository file contents do not match, wherein the synchronizing of the client files and the ~~server~~repository files further includes marking ~~un-matching~~ those unmatched files of the client files and the ~~server~~repository files to be copied to a repository ~~for matching~~ to be synchronized at a later time;

perform a post-synchronization match of the client message digests with the ~~server~~repository message digests and, if the client message digests do not match the ~~server~~repository message digests, detect one or more client files corresponding to one or more unmatched client message digests, and tag the one or more client files; and

re-synchronize the client files and the ~~server~~repository files, the re-synchronization including copying the one or more client files to the ~~server~~repository such that the client message digests and the ~~server~~repository message digests are matched.

Claims 21-28 (Cancelled)

29. (Currently Amended) The system of claim 10, wherein the processor is further to add client file contents that are missing on the ~~server~~repository ~~to the server~~.

30. (Currently Amended) The system of claim 10, wherein the processor is further to combine the client message digests into a single client message digest.

31. (Currently Amended) The machine-readable medium of claim 20, wherein the instructions when executed to perform synchronization, further cause the machine to add client file contents that are missing on the ~~server~~repository ~~to the server~~.

32. (Currently Amended) The machine-readable medium of claim 20, wherein the processor is further to combine the client message digests into a single client message digest.